9. Teorie NP-úplnosti

9.1 Logické formule

Logická (booleovská) proměnná je proměnná, která nabývá hodnot 0 (false) a 1 (true).

Logická formule:

(i) konstanty 0 a 1 a každá logická proměnná jsou logickými formulemi,

(ii) jsou-li f, g logické formule, pak je logická formule i výraz $\bar{f}$, $f \wedge g$, $f \vee g$, $f \Rightarrow g$, $f \Leftrightarrow g$, $f \oplus g$.

$\bar{f}$	*negace*	$\bar{f} = 1$	$\Leftrightarrow$	$f = 0$
$f \wedge g$	*konjunkce*	$f \wedge g = 1$	$\Leftrightarrow$	$f = 1$ a současně $g = 1$
$f \vee g$	*disjunkce*	$f \vee g = 1$	$\Leftrightarrow$	alespoň jeden z f, g je roven 1
$f \Rightarrow g$	*implikace*	$f \Rightarrow g = 1$	$\Leftrightarrow$	$\bar{f} = 1$ nebo $g = 1$
$f \Leftrightarrow g$	*ekvivalence*	$f \Leftrightarrow g = 1$	$\Leftrightarrow$	$f = g$
$f \oplus g$	*vylučovací nebo*	$f \oplus g = 1$	$\Leftrightarrow$	právě jeden z f, g je roven 1

Definice 9.1. *Má-li formule pro dané hodnoty proměnných hodnotu 1, říkáme, že je splněna. Formule, která je splněna pro všechny hodnoty proměnných, se nazývá* tautologie.

Řekneme, že formule f proměnných $x_1, x_2, \ldots, x_n$ je splnitelná, *jestliže existují hodnoty $x_1, x_2, \ldots, x_n$, pro které je f splněna.*

Definice 9.2. *Proměnné a jejich negace se nazývají* literály. *Literály a disjunkce dvou či více literálů se nazývají* (disjunktivní) klauzule.

Je-li formule disjunktivní klauzulí nebo konjunkcí dvou či více disjunktivních klauzulí, říkáme, že formule je v konjunktivní normální formě (tvaru).

Je-li formule v konjunktivní normální formě a každá klauzule obsahuje literály všech proměnných, říkáme, že formule je v úplné konjunktivní normální formě (tvaru).

Definice 9.3. *Literály a konjunkce dvou či více literálů se nazývají* (konjunktivní) klauzule.

Je-li formule konjunktivní klauzulí nebo disjunkcí dvou či více konjunktivních klauzulí, říkáme, že formule je v disjunktivním normálním tvaru (formě).

Věta 9.1. *Každou nekonstantní logickou formuli lze vyjádřit v ÚDNF i ÚKNF.*

9.2 Problém splnitelnosti logických formulí – SAT

SAT

Vstup: logická formule $f(x_1, x_2, \ldots, x_n)$ v proměnných $x_1, x_2, \ldots, x_n$ v KNF (tj. $f = c_1 \wedge c_2 \wedge \ldots \wedge c_m$, kde c_i jsou klauzule proměnných $x_1, x_2, \ldots, x_n$).
Úkol: zjistit, zda je formule f splnitelná.

9.3 Problém k-SAT a polynomialita problému 2-SAT

k-SAT

Vstup: logická formule tvaru

$$f(x_1, x_2, \dots, x_n) = \bigwedge_{i=1}^{m} \left(\bigvee_{j=1}^{k} a_{ij} \right),$$

kde každé a_{ij} je rovno x_ℓ nebo $\bar{x}_\ell$ pro vhodné $\ell = 1, \dots, n$
(tj. f je formule v KNF, která má m klauzulí délky k).

Úkol: zjistit, zda je formule f splnitelná.

2-SAT

Vstup: logická formule $f(x_1, \dots, x_n) = (a_1 \vee b_1) \wedge (a_2 \vee b_2) \wedge \dots, \wedge (a_m \vee b_m)$,
kde každé a_i, $b_i (i = 1, \dots m)$ je rovno x_ℓ nebo $\bar{x}_\ell$ pro vhodné $\ell = 1, \dots, n$
(tj. f je formule v KNF s klauzulemi délky 2).

Úkol: zjistit, zda je formule f splnitelná.

Věta 9.2. 2-SAT $\in$ P.

$f(x_1, \dots, x_n) = C_1 \ \wedge \ C_2 \ \wedge \ \dots \wedge \ C_m$ je formule v proměnných $x_1, \dots, x_n$
s klauzulemi o 2 literálech.

Sestrojíme orientovaný graf $\vec{G}_f$ následující konstrukcí:

$U(\vec{G}_f) = \{x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n\}$,

pro každou klauzuli $C_i = (a \vee b)$ budou v $\vec{G}_f$ obě hrany $(\bar{a}, b)$ a $(\bar{b}, a)$.

Lemma 9.1. *Existuje-li v $\vec{G}_f$ orientovaný sled z a do b, pak v $\vec{G}_f$ existuje i orientovaný sled z $\bar{b}$ do $\bar{a}$.*

Splňující přiřazení: *každý vektor $t \in \{0,1\}^n$, pro který $f(t) = 1$*
(tj. takové přiřazení hodnot 0, 1 proměnným $x_1, \ldots, x_n$, že pro tyto hodnoty je formule f splněna).

Je-li a literál a $t \in \{0,1\}^n$ vektor hodnot proměnných $x_1, \ldots, x_n$, pak $a(t)$ značí hodnotu literálu a po dosazení vektoru t.

Lemma 9.2. *Existuje-li v $\vec{G}_f$ cesta z a do b, pak pro každé splňující přiřazení t je $a(t) = 1 \;\Rightarrow\; b(t) = 1$.*

Lemma 9.3. *Je-li t splňující přiřazení, pak pro každou kvazikomponentu $\vec{G}_i$ grafu $\vec{G}_f$ a pro každé uzly $a, b \in U(\vec{G}_i)$ je $a(t) = b(t)$ (a tedy také $\bar{a}(t) = \bar{b}(t)$).*

Lemma 9.4. *Formule f je splnitelná právě když žádná kvazikomponenta grafu $\vec{G}_f$ neobsahuje současně některou proměnnou i její negaci.*

$\vec{G}_1, \ldots, \vec{G}_s$ – kvazikomponenty $\vec{G}_f$ v acyklickém očíslování
(tj. $\vec{G}_1$ je vstupní, $\vec{G}_s$ je výstupní a hrany z G_i do G_j existují pouze pro $i < j$)
Konstrukce přiřazení t
Postupujeme pro $j = s, s-1, \ldots, 1$,
$\forall$ proměnnou x_i určíme největší index j_0 takový, že x_i nebo $\bar{x}_i$ je v $\vec{G}_{j_0}$, položíme

$x_i = 1$, jestliže $x_i \in U(\vec{G}_{j_0})$,
$x_i = 0$, jestliže $\bar{x}_i \in U(\vec{G}_{j_0})$.

9.4 Problém existence nezávislé množiny uzlů dané velikosti - IND

IND

Vstup: neorientovaný graf G na n uzlech a přirozené číslo $k \leq n$.

Úkol: zjistit, zda v grafu G existuje nezávislá množina uzlů velikosti alespoň k.

Příklad. *Je dán graf G s uzly $x_1, x_2, \ldots, x_n$ a číslo k.*
Sestrojíme formuli $f(u_1, u_2, \ldots, u_n)$ takovou, že f je splněna právě když množina $N = \{x_i \in U(G) | u_i = \mathrm{TRUE}\}$ je nezávislá množina o alespoň k prvcích.

Našli jsme *polynomiální redukci IND na SAT.*

Důsledek 9.1. *Úloha SAT je alespoň tak těžká jako úloha IND.*

Důsledek 9.2. *Kdybychom měli polynomiální algoritmus na SAT, bylo by možné vytvořit polynomiální algoritmus i na IND.*

Příklad. *Naopak: je dána logická formule v KNF tvaru*

$$f(u_1, u_2, \ldots, u_n) = \bigwedge_{i=1}^{m} \left(\bigvee_{j=1}^{m_i} a_{ij} \right),$$

kde každé a_{ij} je tvaru u_k nebo $\bar{u}_k$ pro vhodné k.
Sestrojíme graf G takový, že v G existuje nezávislá množina velikosti m právě když f je splnitelná.

Konstrukce grafu G:
$U(G) = \{x_{ij} |\ i = 1, \ldots, m\,;\ j = 1, \ldots, m_i\}$,
$H(G) = \{\{x_{ij}, x_{pq}\} |\ i = p \text{ a } j \neq q\} \cup \{\{x_{ij}, x_{pq}\} |\ i \neq p,\ j = q \text{ a } a_{ij} = \bar{a}_{pq}\}$.

Našli jsme *polynomiální redukci SAT na IND.*

Důsledek 9.3. *Úloha IND je alespoň tak těžká jako úloha SAT.*

Důsledek 9.4. *Kdybychom měli polynomiální algoritmus na IND, bylo by možné vytvořit polynomiální algoritmus i na SAT.*

Důsledek 9.5. $SAT \in P$ *právě když* $IND \in P$.